

Phylogenies without Branch Bounds: Contracting the Short, Pruning the Deep

Constantinos Daskalakis Elchanan Mossel* Sebastien Roch

July 28, 2009

Abstract

We introduce a new phylogenetic reconstruction algorithm which, unlike most previous rigorous inference techniques, does not rely on assumptions regarding the branch lengths or the depth of the tree. The algorithm returns a forest which is guaranteed to contain all edges that are: 1) sufficiently long and 2) sufficiently close to the leaves. How much of the true tree is recovered depends on the sequence length provided. The algorithm is distance-based and runs in polynomial time.

1 Introduction

In Evolutionary Biology, the speciation history of a family of related organisms is generally represented graphically by a *phylogeny*, that is, a tree where the leaves are the observed (extant) species and the branchings indicate speciation events. Traditional approaches for reconstructing phylogenies from homologous molecular sequences extracted from the observed species [Fel04, SS03] are typically computationally intractable [GF82, DS86, Day87, CT06, Roc06], statistically inconsistent [Fel78], or they require impractical sequence lengths [Att99, LC06, SS99, SS02]. Nevertheless, over the past decade, much progress has been made in the design of efficient, fast-converging reconstruction techniques, starting with the seminal work of Erdős et al. [ESSW99a]. The algorithm in [ESSW99a], often dubbed the Short Quartet Method (SQM), is based on well-known distance-matrix

*E.M. is supported by an Alfred Sloan fellowship in Mathematics and by NSF grants DMS-0528488, and DMS-0548249 (CAREER) and by ONR grant N0014-07-1-05-06.

techniques, that is, it relies on estimates of the evolutionary distance between each pair of species (roughly the time elapsed since their most recent common ancestor). However, unlike other popular distance methods such as Neighbor-Joining [SN87], the key behind SQM’s performance is that it discards long evolutionary distances, whose estimates from sequence comparisons are known to be statistically unreliable. The algorithm works by first building subtrees of small diameter and, in a second stage, glueing the pieces back together.

The Short Quartet Method is in fact guaranteed to return the correct topology from polynomial-length sequences in polynomial time with high probability. But this appealing theoretical performance comes at a price. The results of [ESSW99a] rely critically on biological assumptions which, although reasonable, are often not met in practice (see Section 1.3 for a formal statement):

- a) [*Dense Sampling of Species*] The observed species are “closely related.” In particular, there are no exceptionally long branches in the phylogeny.
- b) [*Absence of Polytomies*] The phylogeny is bifurcating. In fact, Erdős et al. assume that speciation events are sufficiently far apart to be easily distinguished.

The point of a) is that it implies a natural bound on the depth of the tree which in turn ensures that enough information about the deep parts of the tree diffuses to the leaves. As for Assumption b), it guarantees that a clear signal can be extracted from each branch of the phylogeny. It is obvious—at least intuitively—that assumptions such as a) and b) are necessary to secure the type of results Erdős et al. obtain: *the guaranteed reconstruction of the full phylogeny*. Hence, to improve over SQM and obtain strong guarantees under more general conditions, one has to relax this last requirement.

In this paper, we design an algorithm which provides strong reconstruction guarantees without Assumptions a) and b). We show that our algorithm is guaranteed to recover a *forest* containing all edges that are “sufficiently long” and “sufficiently close” to the leaves. In fact, we allow a trade-off between the resolution of short branches and the depth of the reconstructed forest, a feature of potential practical interest. Also, we guarantee that our reconstructed forest has the desirable property of being *disjoint* (although the presence of short edges leads us to allow deep intersections of very short branches between the subtrees). Moreover, our algorithm does not require the knowledge of a priori bounds on branch lengths or tree depth. Finally

if Assumptions a) and b) are satisfied, we recover the whole phylogeny and provide an alternative to the algorithm of Erdős et al.

Precise statements are given in Section 1.2. For a full comparison to related work see also Section 1.3.

1.1 What can we hope to reconstruct?

Well-known identifiability results [Cha96] guarantee that phylogenies—or at least their idealized stochastic models—can be fully reconstructed given enough data at the leaves. However, molecular data gathered from current species are in essence limited, which begs the question: *How much of the tree can we really hope to reconstruct?* We pointed out above two important sources of difficulties: short branches produce a weak signal that may be hard to detect; similarly, untangling the deep parts of the tree presents challenges that are well documented (see, e.g., [PL98, CDvM⁺06]). Note that these issues are fundamentally “information-theoretic” and affect all reconstruction methods.

To avoid these difficulties, most *rigorous* methods impose restrictions on the length of the branches and/or the depth of the tree, which may be unsatisfactory from a practical perspective. On the other hand, methods commonly used in *practice*, such as likelihood and bayesian methods, typically produce several candidate trees as well as confidence estimates. But theoretical guarantees on the quality of such outputs are hard to obtain.

Here, we seek to give strong reconstruction guarantees without any assumption on the true phylogeny. Our goal is to recover, for any given amount of data, as much of the tree as can rigorously be reconstructed with high confidence. Since the full phylogeny may not always be recoverable, we are led to a more flexible solution concept: we output a *contracted subforest* of the true phylogeny. That is, we output a forest containing all branches that are “sufficiently long” and “sufficiently recent”; note that “sufficiently” here is determined (information-theoretically) by the size of the data (usually in terms of sequence length). In the remainder of this section we formalize this solution concept.

The input. Formally, a phylogeny is a *weighted, multifurcating tree* on a set of leaves L , which we identify with the labels $[n] = \{1, \dots, n\}$. We denote a phylogeny by $T = (V, E; L, \lambda)$. Here V and E are respectively the vertex and edge set of the tree, and $\lambda : E \rightarrow (0, +\infty)$ assigns a weight to each edge (the branch length). We assume that all internal vertices $V - L$ have degree at least 3.

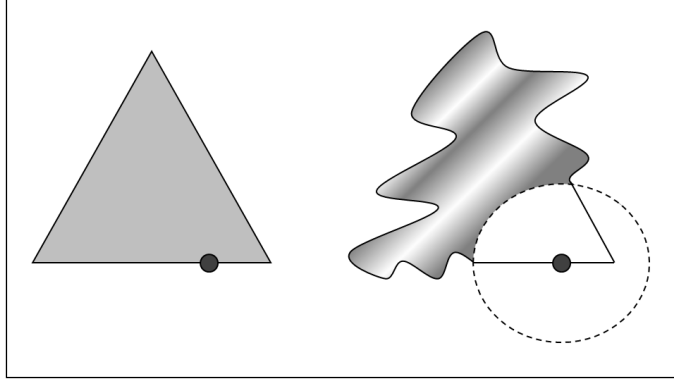


Figure 1: The effect of distance distortion from the perspective of a leaf. On the left hand side is the true phylogeny. On the right hand side, only distances within a certain radius represent accurately the metric underlying the phylogeny.

A phylogeny is naturally equipped with a so-called additive metric on the leaves $d : L \times L \rightarrow (0, +\infty)$ defined as follows

$$\forall u, v \in L, d(u, v) = \sum_{e \in P_T(u, v)} \lambda_e,$$

where $P_T(u, v)$ is the set of edges on the path between u and v in T . Often $d(u, v)$ is referred to as the “evolutionary distance” between species u and v . Since under the assumptions above there is a one-to-one correspondence between d and λ , we write either $T = (V, E; L, d)$ or $T = (V, E; L, \lambda)$. We also sometimes use the natural extension of d to the internal vertices of T . We denote by \mathcal{T} the set of all phylogenies on any number of leaves.

It is well-known that given an additive metric d one can reconstruct the corresponding phylogeny T . However, in practice, one can only derive an *estimate* \hat{d} of d , the accuracy of which depends on the amount of data used. (This estimate is known in the literature as the “distance matrix”.) Our goal in this paper is to reconstruct a phylogeny—or as much of it as possible—from this “distorted” version of its additive metric. A typical property of distance estimates is that estimates of long distances are unreliable. The following definition formalizes this phenomenon. See Figure 1 for an illustration.

Definition 1 (Distorted Metric [Mos07, KZZ03]) *Let $T = (V, E; L, d)$ be a phylogeny and let $\tau, M > 0$. We say that $\hat{d} : L \times L \rightarrow (0, +\infty]$ is a (τ, M) -distorted metric for T or a (τ, M) -distortion of d if:*

1. [Symmetry] For all $u, v \in L$, \hat{d} is symmetric, that is,

$$\hat{d}(u, v) = \hat{d}(v, u);$$

2. [Distortion] \hat{d} is accurate on “short” distances, that is, for all $u, v \in L$, if either $d(u, v) < M + \tau$ or $\hat{d}(u, v) < M + \tau$ then

$$\left| d(u, v) - \hat{d}(u, v) \right| < \tau.$$

In phylogenetic reconstruction, a distorted metric is naturally derived from samples of a Markov model on a tree—a common model of DNA sequence evolution used in Biology. (See Appendix A for details.) In the remainder of this paper, we assume that we are given a (τ, M) -distortion \hat{d} of an additive metric d and we seek to recover the underlying phylogeny T .

Contraction and pruning. Given only a (τ, M) -distorted metric, it is clear that the best we can hope for in general is to reconstruct a forest containing those edges of T that are “sufficiently close” to the leaves. Indeed, note that two phylogenies that are identical up to depth M from the leaves, but are otherwise different, can give rise to the same distorted metric. Moreover, since we do not assume that edges are longer than the accuracy τ , some edges may be too short to be reconstructed and, as we mentioned before, we allow ourselves to instead contract them. Hence, we are led to consider subforests of the true phylogeny where deep edges are *pruned* and short edges are *contracted*.

To formalize this idea we need a few definitions. Let us first describe what we mean by a *subforest* of a phylogeny $T = (V, E; L, d)$. Given a set of vertices $V' \subseteq V$, the *subtree of T restricted to V'* is the tree obtained 1) by keeping only nodes and edges on paths between vertices in V' and then 2) by contracting all paths composed of vertices of degree 2, except the nodes in V' . See Figure 2 for an example. We denote this tree by $T|_{V'}$. We typically take $V' \subseteq L$. A *subforest* of T is defined to be a collection of restricted subtrees of T .

We also need a notion of depth. Given an edge $e \in E$, the *chord depth* of e is the length of the shortest path among all paths crossing e between two leaves. That is,

$$\Delta_c(e) = \min \{ d(u, v) : u, v \in L, e \in P_T(u, v) \}.$$

We define the *chord depth of a tree T* to be the maximum chord depth in T

$$\Delta_c(T) = \max \{ \Delta_c(e) : e \in E \}.$$

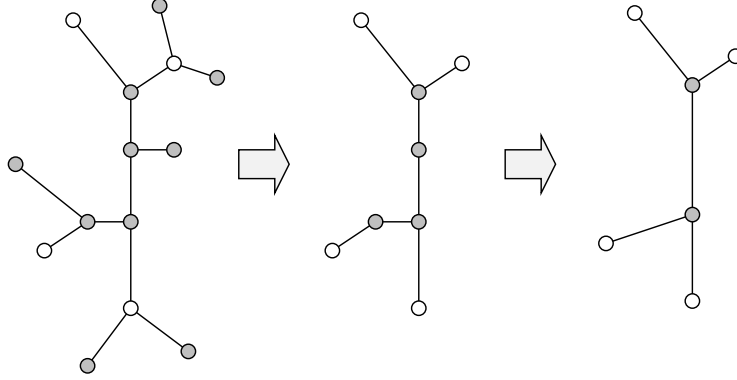


Figure 2: Restricting the top tree to its white nodes.

Definition 2 (Contracted Subforest) Let $T = (V, E; L, d)$ be a phylogeny. Fix $M > 0$. Let $\{L_1, \dots, L_q\}$ be the natural partition of the leaf set L obtained by removing all edges $e \in E$ such that $\Delta_c(e) \geq M$. We define the M -pruned subforest of T to be the forest $F_M(T) = (V_M, E_M)$ consisting of the trees $\{T|_{L_1}, \dots, T|_{L_q}\}$. The metric d is extended as follows for all $u, v \in L$,

$$d_M(u, v) = \begin{cases} d(u, v), & \text{if } u, v \text{ are in the same subtree of } F_M(T), \\ +\infty, & \text{o.w.} \end{cases}$$

We also denote by λ_M the edge lengths of $F_M(T)$.

Now, given also $\tau > 0$, the τ -contracted M -pruned subforest of T is the forest $F_{\tau, M}(T) = (V_{\tau, M}, E_{\tau, M})$ obtained from $F_M(T)$ by contracting edges $e \in E_M$ of weight $\lambda_M(e) \leq \tau$.

Path-disjointness. We require that the trees of our reconstructed forest are “non-intersecting”. This is a natural condition to impose in order to obtain a meaningful reconstruction: we want to avoid as much as possible that the same branches appear in many subtrees. In fact, we can only guarantee approximate disjointness as defined below.

We first need a notion of depth for vertices. For a phylogeny $T = (V, E; L, d)$ and a vertex $x \in V$, the *vertex depth* of x is the length of the shortest path between x and the set of leaves. That is,

$$\Delta_v(x) = \min \{d(u, x) : u \in L\}.$$

Given two leaves u, v of T , we denote by $\tilde{P}_T(u, v)$ the set of vertices on the path between u and v in T .

We say that two trees are (τ, M) -path disjoint if they are “almost disjoint” in the sense that they only share edges (if any) that are “deep” (end-points have vertex depth at least $M/2$) and “short” (length at most τ). More formally:

Definition 3 (Approximate Path-Disjointness) *Let $T = (V, E; L, d)$ be a phylogeny. Two subtrees T_1, T_2 of T restricted respectively to $L_1, L_2 \subseteq L$ are (τ, M) -path-disjoint if $L_1 \cap L_2 = \emptyset$ and for all pairs of leaves $u_1, v_1 \in L_1$ and $u_2, v_2 \in L_2$ such that*

$$\tilde{P}_T(u_1, v_1) \cap \tilde{P}_T(u_2, v_2) \neq \emptyset,$$

we have:

$$\min\{\Delta_v(x) : x \in \tilde{P}_T(u_1, v_1) \cap \tilde{P}_T(u_2, v_2)\} \geq \frac{1}{2}M,$$

and, if further $P_T(u_1, v_1) \cap P_T(u_2, v_2) \neq \emptyset$,

$$\max\{\lambda_e : e \in P_T(u_1, v_1) \cap P_T(u_2, v_2)\} \leq \tau.$$

More generally, a collection of restricted subtrees T_1, \dots, T_q of T are (τ, M) -path-disjoint if they are pairwise (τ, M) -path-disjoint. In the case $\tau = 0$, we simply say that the subtrees are path-disjoint.

1.2 Main result and corollaries

Main result. Our main result is an algorithm which, given a (τ, M) -distorted metric, reconstructs a contracted subforest (of the true phylogeny) whose trees are approximately path-disjoint. Typically, M is much larger than τ . In that case, we reconstruct a subforest of T with chord depth $\approx \frac{1}{2}M$ which includes all edges of length at least 4τ . The reconstructed subtrees may “overlap” on edges of length at most 2τ at vertex depth $\gtrsim \frac{1}{4}M$. In Section 4, we show that these parameters are essentially optimal. The algorithm runs in polynomial time.

More precisely, we show:

Theorem 1 (Main Result) *Let τ and M be monotone functions of n with $M > 3\tau$. Let $m > 3\tau$ be such that*

$$m < \frac{1}{2}[M - 3\tau],$$

for all n . Then, there is a polynomial-time algorithm \mathcal{A} such that, for all phylogenies $T = (V, E; L, d)$ in \mathcal{T} with $|L| = n$ and all (τ, M) -distortions \hat{d} of d , \mathcal{A} applied to \hat{d} satisfies the following:

1. [Approximate Path Disjointness] \mathcal{A} returns a $(2\tau, m-3\tau)$ -path-disjoint subforest \hat{F} of T ;
2. [Depth Guarantee] The forest \hat{F} is a refinement of $F_{4\tau, m-\tau}(T)$;

We give below a few important special cases of Theorem 1.

Tree case. When the amount of data is sufficient to produce a distorted metric with $M = \Omega(\Delta_c(T))$, we get a single component, that is, the full tree (up to those edges that are contracted).

Corollary 1 (Tree Case) *Let $\tau > 0$ and $M > 2\Delta_c(T) + 5\tau$. Then, choosing $m > \Delta_c(T) + \tau$ guarantees that the reconstructed forest is composed of only a tree.*

In the case of “dense” phylogenies, $M = \Omega(\log n)$ is sufficient to reconstruct the full tree.

Definition 4 (Dense Phylogenies (see e.g. [ESSW99a])) *We say that a collection of phylogenies \mathcal{T}' is dense if there is a $0 < g < +\infty$ (independent of n) such that for all $T = (V, E; L, \lambda) \in \mathcal{T}'$ we have*

$$\forall e \in E, \lambda_e \leq g. \tag{1}$$

We denote by \mathcal{T}_g the set of phylogenies satisfying (1).

Corollary 2 (Dense Case) *In the case of dense phylogenies, $M = \Omega(\log n)$ suffices to guarantee the reconstruction of the full tree, up to contracted edges.*

Absolute variant. All rigorous algorithms prior to our work (see Section 1.3) require knowledge of either the tree depth or bounds on the edge lengths to give strong reconstruction guarantees. This is not satisfactory from a practical point of view. Here given only the sequence length we provide explicit guarantees. The following result assumes that the distorted metric is derived from a Markov model on a tree. (See Appendix A for details.)

Corollary 3 (Absolute Variant) *Given a number of samples $k = \Omega(\log n)$ from a Markov model on a tree and a chosen level of contraction $\varepsilon > 0$ (small), one can choose τ, M, m so that \mathcal{A} is guaranteed to return a (contracted) subforest of T containing $F_{\varepsilon, M'}(T)$ with probability $1 - o(1)$, where $M' = \Omega_\varepsilon(\log k - \log \log n)$.*

Complete resolution. Finally we remark that, if we further assume that all branch lengths are bounded *from below* by a constant, then by choosing τ accordingly a non-contracted forest is returned. In particular, we can recover the results of [ESSW99a].

1.3 Related work

Under a Markov model of evolution, the Short Quartet Method (SQM) of Erdős et al. [ESSW99a] is guaranteed to recover the full phylogeny as long as the number of samples k satisfies

$$k > cf^{-2}e^{c'g\Delta_c(T)} \log n,$$

for constants $c, c' > 0$, where f and g are respectively lower and upper bounds on the branch lengths possibly depending on n . For instance, if f and g are constants the sequence length needed for complete reconstruction depends polynomially in the number of species.

Mossel [Mos07] developed a framework that allows the reconstruction of a well-behaved *forest* when sequences are too short to guarantee a complete reconstruction. More precisely, edges which are too deep (in the sense of appearing only on paths between species whose distances are not accurately known) are *pruned* from the final reconstruction. At a high level, Mossel’s Distorted Metric Method (DMM) (implicit in [Mos07]), works in a fashion similar to SQM—except for a pre-processing phase that clusters together sufficiently related species. However, for DMM to work, lower bounds on the branch lengths are required and, moreover, these must be known by the algorithm. Following up on [Mos07], Daskalakis et al. [DHJ⁺06] gave a variant of DMM that runs without knowledge of a priori bounds on the branch lengths or the tree depth—making their variant somewhat more practical. However, like DMM, the algorithm in [DHJ⁺06] does not deal properly with short edges: any part of the tree containing a short edge cannot be reconstructed by the algorithm (even though there may be adjacent edges that are in fact reconstructible). Therefore, in the presence of short edges no guarantee can be given about the depth of the reconstructed forest.

	<i>Execution</i>	<i>Guarantees</i>	
	No branch bound needed	Short edges OK	Deep edges OK
[ESSW99a]			
[Mos07]			✓
[DHJ ⁺ 06]	✓		✓
[GMS08]	✓	✓	
Our method	✓	✓	✓

Figure 3: Comparison of methods.

Recently Gronau et al. [GMS08] eliminated the need for a lower bound on the branch length by *contracting* edges whose length is below a user-defined threshold. Their solution uses a Directional Oracle (DO) which closes in on the location of a leaf to be added and, in the process, contracts regions that do not provide a reliable directional signal. Although the DO algorithm does not use an explicit bound on the depth of the tree, their *reconstruction guarantee* requires such a bound, similarly to [ESSW99a]. In particular, Gronau et al. leave open the question of giving a forest-building version of their algorithm. Moreover, the sequence length in [GMS08] depends exponentially on what the authors call the ε -diameter of the tree—essentially, the maximum diameter of the contracted regions. It is natural to conjecture that an optimal result should not depend on this parameter.

For further related work on efficient phylogeny reconstruction, see also [ESSW99b, HNW99, CK01, Csu02, KZZ03, MR06, DMR06].

1.4 Discussion of the results

In Table 3 we summarize the current status as discussed in the previous sections.

As the table emphasizes, our overarching goal is to design an algorithm with good reconstruction guarantees in the presence of both short and deep edges, whose execution does not rely on a priori bounds on branch lengths. Unfortunately, given the combinatorial complexity of Mossel’s forest-building algorithm, it is not straightforward to provide the extra flexibility of edge contraction in this framework. The novelty in our work is twofold:

- *Solution Concept:* A basic complication is that, in some sense, contraction and pruning interfere with each other. Indeed, the presence

of unresolved branches at the boundary of partially reconstructed subtrees creates the possibility of deep “undetectable” intersections. This pitfall seems to be unavoidable. One of our main contributions is to introduce the notion of approximate disjointness, which allows short but deep intersections between subtrees of the reconstructed forest. This suitable solution concept leads to a quite simple algorithm with reasonable guarantees. Moreover, the flexibility in our definition allows us to recover all previously known results as special cases.

- *Algorithmic Technique:* A natural approach to forest building used in [Mos07, DHJ⁺06] proceeds along the following three steps:
 1. first, leaves are grouped into clusters for which all pairwise distances are accurately known (the *small* clusters);
 2. by definition, the local topologies on the small clusters can be trivially reconstructed [Bun71];
 3. finally, the local topologies that intersect in the true tree are “glued” together to get a forest (the resulting forest partitions the leaves into *large* clusters).

This last step involves non-trivial combinatorial considerations. We have found that further allowing contracted edges makes this process somewhat unmanageable. Instead we use a different approach relying on simple metric arguments. In particular, we *directly* partition the leaves into large clusters, whose underlying subtrees are approximately disjoint, and provide a new straightforward method to reconstruct these subtrees.

In addition, we obtain as special cases the results discussed in Section 1.3. In particular, if there are no short edges, we recover the results of [Mos07] and [DHJ⁺06], where a path-disjoint forest is returned (by taking τ equal to half the lower bound on the branch lengths in Theorem 1). If furthermore there is an upper bound on the branch lengths, we recover the results of [ESSW99a] (Corollary 2). Finally, if we keep the upper bound on the edge lengths, but drop the lower bound, we recover the results of [GMS08] (Corollary 1). In fact, we eliminate the dependence on the ε -diameter.¹ Further,

¹After the results of the current paper were posted on the arXiv, we were informed by S. Moran that, in parallel to our work, the authors of [GMS08] have improved on their previous results: the dependence on the ε -diameter has been removed. A preprint of this work is currently available on the authors’ website. Note however that this new, independent work does not deal with deep edges and still makes assumptions similar to [ESSW99a] restricting the depth of the generating tree.

unlike [GMS08], we allow an arbitrary number of states, an extension—it should be noted—that follows easily from [ESSW99b] and [Mos07].

1.5 Organization

The rest of the paper is organized as follows. The algorithm is detailed in Section 2. The proof of our main theorem follows in Section 3. We conclude with a lower bound in Section 4 and a discussion of the running time in Section 5. Also, for completeness, in Appendix A we describe the probabilistic motivation behind the distorted metric definition.

The results in this paper were announced without proof in [DMR09]. Also, the counter-example in Section 4 did not appear in [DMR09].

2 Algorithm

The outline of the algorithm follows. There are three main phases, which are explained in detail after the outline. The input to the algorithm is a (τ, M) -distorted metric \hat{d} on n leaves. In particular, we assume that the values τ and M are known to the algorithm (but see also Corollary 3). Let m be as in Theorem 1. We denote the true tree by $T = (V, E; L, d)$. The details of the subroutines MINI CONTRACTOR and EXTENDER are detailed in Figures 5 and 7 (see also their high level description below).

- **Pre-Processing: Leaf Clustering.** Build the distorted clustering graph $\hat{H}_m = (\hat{V}_m, \hat{E}_m)$ where $\hat{V}_m = [n]$ and $(u, v) \in \hat{E}_m \iff \hat{d}(u, v) < m$; compute the connected components $\{\hat{h}_m^{(i)} = (\hat{v}_m^{(i)}, \hat{e}_m^{(i)})\}_{i=1}^q$ of \hat{H}_m ;
- **Main Loop.** For all components $i = 1, \dots, q$:
 - For all pairs of leaves $u, v \in \hat{v}_m^{(i)}$ such that $(u, v) \in \hat{E}_m$:
 - * **Mini Reconstruction.** Compute

$$\{\psi_j(u, v)\}_{j=1}^{r(u,v)} := \text{MINI CONTRACTOR}(\hat{h}_m^{(i)}; u, v);$$
 - * **Bipartition Extension.** Compute

$$\{\bar{\psi}_j(u, v)\}_{j=1}^{r(u,v)} := \text{EXTENDER}(\hat{h}_m^{(i)}, \{\psi_j(u, v)\}_{j=1}^{r(u,v)}; u, v);$$
 - Deduce the tree $\hat{T}^{(i)}$ from $\{\bar{\psi}_j(u, v)\}_{j=1}^{r(u,v)}$;
- **Output.** Return the resulting forest \hat{F} .

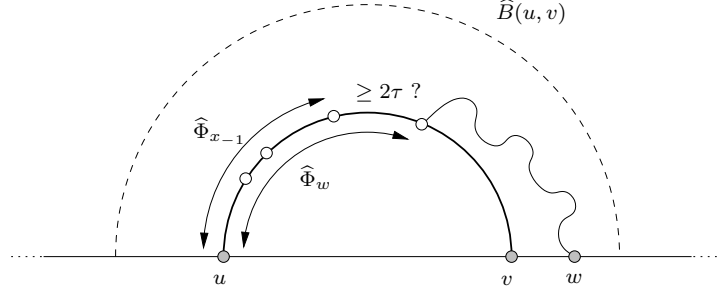


Figure 4: Illustration of routine MINI CONTRACTOR. See Figure 5 for notation.

Pre-processing: Leaf clustering. As mentioned before, given a (τ, M) -distortion we cannot hope to reconstruct edges that are too deep inside the tree. This results in the reconstruction of a *forest*. Therefore, the first phase of the algorithm is to determine the “support” of this forest. We proceed as follows. Consider the following graph on L .

Definition 5 (Clustering Graph) Let $M' \in [\tau, \leq M - \tau]$. The distorted clustering graph with parameter M' , denoted $\hat{H}_{M'} = (\hat{V}_{M'}, \hat{E}_{M'})$, is the following graph: the vertices $\hat{V}_{M'}$ are the leaves L of T ; two leaves $u, v \in L$ are connected by an edge $e = (u, v) \in \hat{E}_{M'}$ if

$$\hat{d}(u, v) < M'. \quad (2)$$

Note that this is an undirected graph because \hat{d} is symmetric. Similarly, we define the clustering graph with parameter M' , $H_{M'} = (V_{M'}, E_{M'})$, where we use d instead \hat{d} in (2).

The first phase of the algorithm consists in building the graph \hat{H}_m from \hat{d} . We then compute the connected components of \hat{H}_m which we denote $\{\hat{h}_m^{(i)}\}_{i=1}^q$. In the next two phases, we build a tree on each of these components.

Building the components I: Mini-reconstruction problem. Fix a component $\hat{h}_m^{(i)}$ of \hat{H}_m . In this and the next phase, we seek to reconstruct a contracted tree on $\hat{h}_m^{(i)}$. Denote by $T^{(i)}$ the true tree T restricted to the leaves in $\hat{h}_m^{(i)}$. First, we find all edges of $T^{(i)}$ that are “sufficiently long” and lie on “sufficiently short” paths. More precisely, we consider all pairs

Algorithm MINI CONTRACTOR*Input:* Component $\hat{h}_m^{(i)}$; Leaves u, v ;*Output:* Bipartitions $\{\psi_j(u, v)\}_{j=1}^{r(u, v)}$;

- **Ball.** Let

$$\hat{B}(u, v) := \left\{ w \in \hat{v}_m^{(i)} : \hat{d}(u, w) \vee \hat{d}(v, w) < M \right\};$$

- **Intersection Points.** For all $w \in \hat{B}(u, v)$, estimate the point of intersection between u, v, w (distance from u), that is,

$$\hat{\Phi}_w := \frac{1}{2} \left(\hat{d}(u, v) + \hat{d}(u, w) - \hat{d}(v, w) \right);$$

- **Long Edges.** Set $S := \hat{B}(u, v) - \{u\}$, $x_{-1} = u$, $j := 0$, $C_0 = \{u\}$;

- Until $S = \emptyset$:

- * Let $x_0 = \arg \min \{\hat{\Phi}_w : w \in S\}$ (break ties arbitrarily);
- * If $\hat{\Phi}_{x_0} - \hat{\Phi}_{x_{-1}} \geq 2\tau$, create a new edge by setting $\psi_{j+1}(u, v) := \{\hat{B}(u, v) - S, S\}$ and let $C_{j+1} := \{x_0\}$, $j := j + 1$;
- * Else, set $C_j := C_j \cup \{x_0\}$;
- * Set $S := S - \{x_0\}$, $x_{-1} := x_0$;

- **Output.** Return the bipartitions $\{\psi_j(u, v)\}_{j=1}^{r(u, v)}$ (where $r(u, v)$ is the number of bipartitions generated in the previous step).

Figure 5: Algorithm MINI CONTRACTOR. See Figure 4 for illustration.

of leaves u, v connected by an edge in $\hat{h}_m^{(i)}$, that is, leaves within distorted distance m . For each such pair u, v , the *mini reconstruction problem* consists in finding all edges e in $P_{T^{(i)}}(u, v)$ that have length larger than $\lambda_e \geq 4\tau$. To do this using the distortion \hat{d} , we first consider a ball $\hat{B}(u, v)$ of all nodes within distorted distance M of u and v , that is,

$$\hat{B}(u, v) = \left\{ w \in \hat{h}_m^{(i)} : \hat{d}(u, w) \vee \hat{d}(v, w) < M \right\},$$

where $a \vee b$ is the maximum of a and b . The point of using this ball is that we can then guarantee that each edge in $P_{T^{(i)}}(u, v)$ is “witnessed” by a quartet (i.e., a 4-tuple of leaves) in $\hat{B}(u, v)$ in the following sense: let (x_1, x_2) be an edge in $P_{T^{(i)}}(u, v)$ and let (x_j, y_j) , $j = 1, 2$, be an edge adjacent to x_j that is *not* in $P_{T^{(i)}}(u, v)$; for $j = 1, 2$ let $L_{x_j \rightarrow y_j}^{(i)}$ be the leaves reachable from y_j using paths not including x_j ; then we will show that $L_{x_j \rightarrow y_j}^{(i)} \cap \hat{B}(u, v) \neq \emptyset$ for $j = 1, 2$. In other words, there is enough information in $\hat{B}(u, v)$ to reconstruct all edges in $P_{T^{(i)}}(u, v)$ —at least those that are “sufficiently long.” This phase is detailed in Figure 5. An illustration is given in Figure 4.

Building the components II: Extending the bipartitions. The previous step reconstructs “sufficiently long” edges on balls of the form $\hat{B}(u, v)$. By *reconstructing an edge on $\hat{B}(u, v)$* , we mean *finding the bipartition of $\hat{B}(u, v)$ to which the edge corresponds*. More precisely:

Definition 6 (Bipartitions) *Let $T = (V, E)$ be a multifurcating tree with no vertex of degree 2. Each edge e in T induces a bipartition of the leaves L of T as follows: if one removes the edge e from T , then one is left with two connected components; take the partition of the leaves corresponding to those components. Denote by $b_T(e)$ the bipartition of e on T . It is easy to see that given the bipartitions $\{b_T(e)\}_{e \in E}$ one can reconstruct the tree T efficiently [Bun71, Mea81, BD86]. (Proceed by sequentially “splitting” clusters.)*

The goal of the second phase in the main loop of our reconstruction algorithm is to *extend* the bipartitions previously built from $\hat{B}(u, v)$ to the full component $\hat{h}_m^{(i)}$. To perform this task, we use the following observation: suppose we want to deduce the bipartition corresponding to edge e ; since the radius of the ball $\hat{B}(u, v)$ is much larger than m , we can make sure that a path *from* a leaf in $\hat{h}_m^{(i)}$ that is outside $\hat{B}(u, v)$ *to* a leaf on the other side of the bipartition $b_T(e)$ is “long.” Therefore, we can easily determine what

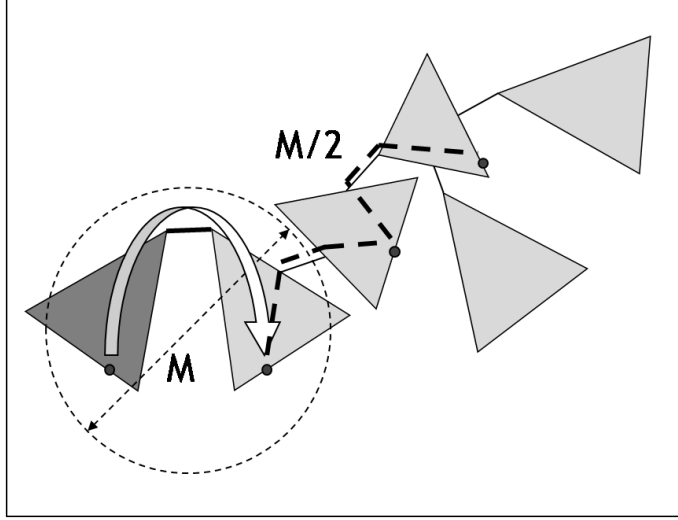


Figure 6: Illustration of routine EXTENDER. See also Figure 7.

side of the partition each leaf in $\hat{h}_m^{(i)}$ lies on. For details, see Figure 7. An illustration is given in Figure 6.

3 Analysis of the Algorithm

We assume throughout that \hat{d} is a (τ, M) -distortion of d and moreover that m satisfies the conditions of Theorem 1.

3.1 Leaf clustering: Determining the support of the forest

Recall the notation of Definition 5.

Proposition 1 (Leaf Clustering) *Let $\tau \leq M' \leq M - \tau$. Then*

$$E_{M'-\tau} \subseteq \hat{E}_{M'} \subseteq E_{M'+\tau}.$$

Proof: This follows immediately from the definition of \hat{d} . Indeed, if $d(u, v) < M' - \tau$ then

$$\hat{d}(u, v) < d(u, v) + \tau < (M' - \tau) + \tau < M'.$$

Similarly, if $\hat{d}(u, v) < M'$ then

$$d(u, v) < \hat{d}(u, v) + \tau < M' + \tau.$$

Algorithm EXTENDER

Input: Component $\hat{h}_m^{(i)}$; Bipartitions $\{\psi_j(u, v)\}_{j=1}^{r(u, v)}$; Leaves u, v ;

Output: Bipartitions $\{\bar{\psi}_j(u, v)\}_{j=1}^{r(u, v)}$;

- For $j = 1, \dots, r(u, v)$ (unless $r(u, v) = 0$):
 - **Initialization.** Denote by $\psi_j^{(u)}(u, v)$ the vertex set containing u in the bipartition $\psi_j(u, v)$, and similarly for v ; Initialize the extended partition $\bar{\psi}_j^{(u)}(u, v) := \psi_j^{(u)}(u, v)$, $\bar{\psi}_j^{(v)}(u, v) := \psi_j^{(v)}(u, v)$;
 - **Modified Graph.** Let K be $\hat{h}_m^{(i)}$ where all edges between $\psi_j^{(u)}(u, v)$ and $\psi_j^{(v)}(u, v)$ have been removed;
 - **Extension.** For all $w \in \hat{v}_m^{(i)} - (\psi_j^{(u)}(u, v) \cup \psi_j^{(v)}(u, v))$, add w to the side of the partition it is connected to in K (by Proposition 6, each w as above is connected to exactly one side);
- Return the bipartitions $\{\bar{\psi}_j(u, v)\}_{j=1}^{r(u, v)}$.

Figure 7: Algorithm EXTENDER. See Figure 6 for an illustration.

■

3.2 Mini-reconstruction: Finding long edges on short paths

Consider a component $\hat{h}_m^{(i)} = (\hat{v}_m^{(i)}, \hat{e}_m^{(i)})$ of \hat{H}_m . Denote by $T^{(i)} = (V^{(i)}, E^{(i)})$ the tree T restricted to the leaves in $\hat{v}_m^{(i)}$, that is,

- Keep only those edges of T that are on paths between leaves in $\hat{v}_m^{(i)}$;
- Glue together edges adjacent to vertices of degree 2;
- Equip $T^{(i)}$ with the metric d restricted to $\hat{v}_m^{(i)} \times \hat{v}_m^{(i)}$ and denote $\{\lambda_e^{(i)}\}_{e \in E^{(i)}}$ the corresponding weights.

Proposition 2 (Chord Depth of $T^{(i)}$) *The chord depth of $T^{(i)}$ is less than $m + \tau$.*

Proof: We argue by contradiction. Let e be an edge in $T^{(i)}$. Suppose that the chord depth of e in $T^{(i)}$ is $\geq m + \tau$. Consider the bipartition $\{\psi^{(1)}, \psi^{(2)}\}$ defined by e in $T^{(i)}$. Then it follows that for all $u_1 \in \psi^{(1)}$ and $u_2 \in \psi^{(2)}$, we have

$$\hat{d}(u_1, u_2) > d(u_1, u_2) - \tau \geq m,$$

so that $\hat{h}_m^{(i)}$ cannot be connected, a contradiction. ■

Let $e' = (u', v')$ be an edge in a tree T' with leaf set L' . We denote by $L'_{u' \rightarrow v'}$ the leaves of T' that can be reached from v' without going through u' . Recall that for two leaves u', v' of T' , we denote by $\tilde{P}_{T'}(u', v')$ the set of vertices on the path between u' and v' in T' . Recall also that

$$\hat{B}(u, v) = \left\{ w \in \hat{v}_m^{(i)} : \hat{d}(u, w) \vee \hat{d}(v, w) < M \right\},$$

for $u, v \in \hat{v}_m^{(i)}$.

Proposition 3 (Witnesses in $\hat{B}(u, v)$) *Assume that $2m + 3\tau < M$. Let $(u, v) \in \hat{e}_m^{(i)}$. Let (x, y) be an edge of $T^{(i)}$ such that $x \in \tilde{P}_{T^{(i)}}(u, v)$ but $y \notin \tilde{P}_{T^{(i)}}(u, v)$. Then we have*

$$\hat{B}(u, v) \cap L_{x \rightarrow y}^{(i)} \neq \emptyset,$$

where $L^{(i)}$ is the set of leaves of $T^{(i)}$.

Proof: By Proposition 2, there are leaves x_0, y_0 in $L^{(i)}$ such that $(x, y) \in P_{T^{(i)}}(x_0, y_0)$ and $d(x_0, y_0) < m + \tau$. Assume without loss of generality that $y_0 \in L_{x \rightarrow y}^{(i)}$. By assumption,

$$d(u, v) < \hat{d}(u, v) + \tau < m + \tau.$$

Therefore,

$$d(u, y_0) \leq d(u, x) + d(x, y_0) \leq d(u, v) + d(x_0, y_0) < 2m + 2\tau,$$

from which we get $\hat{d}(u, y_0) < 2m + 3\tau < M$. The same inequality holds for $\hat{d}(v, y_0)$. ■

Fix a pair of leaves u, v with $(u, v) \in \hat{e}_m^{(i)}$. For $w \in \hat{B}(u, v)$, let

$$\hat{\Phi}_w := \frac{1}{2} \left(\hat{d}(u, v) + \hat{d}(u, w) - \hat{d}(v, w) \right),$$

and

$$\Phi_w := \frac{1}{2} (d(u, v) + d(u, w) - d(v, w)).$$

Note that Φ_w is the distance between u and the intersection point of $\{u, v, w\}$. Let $\{C_j\}_{j=0}^{r(u,v)}$ and $\{\psi_j(u, v)\}_{j=1}^{r(u,v)}$ be as in Figure 5. We write $w \sim w'$ if $w, w' \in C_j$ for some j . Similarly, we write $w \lesssim w'$ (respectively $w < w'$) if $w \in C_j$ and $w' \in C_{j'}$ with $j \leq j'$ (respectively $j < j'$).

Proposition 4 (Intersection Points) *Let u, v be as above. Then we have the following:*

1. [Identity] *If $x, y \in \widehat{B}(u, v)$ are such that $\Phi_x = \Phi_y$ then $x \sim y$;*
2. [Precedence] *If $x, y \in \widehat{B}(u, v)$ are such that $\Phi_x \leq \Phi_y$ then $x \lesssim y$;*
3. [Separation] *If $x, y \in \widehat{B}(u, v)$ are such that $\Phi_x < \Phi_y - 4\tau$ and there is no $z \in \widehat{B}(u, v)$ with $\Phi_x < \Phi_z < \Phi_y$, then $x < y$.*

Proof: For Part 1, note that $\Phi_x = \Phi_y$ implies

$$\left| \widehat{\Phi}_x - \widehat{\Phi}_y \right| < 2\tau.$$

(Note that the term $\hat{d}(u, v)$ appears in both $\widehat{\Phi}_x$ and $\widehat{\Phi}_y$ and therefore does not contribute to the error. The same argument applies to the error calculations below.) Therefore, x and y are necessarily placed in the same C_j , that is, $x \sim y$. See Figure 5.

For Part 2, suppose by contradiction that $x > y$. Then we have necessarily

$$\widehat{\Phi}_x \geq \widehat{\Phi}_y + 2\tau,$$

which implies

$$\Phi_y < \Phi_x - 2\tau + 2\tau \leq \Phi_x,$$

a contradiction.

For Part 3, let

$$X_0 = \{w \in \widehat{B}(u, v) \text{ s.t. } \Phi_w \leq \Phi_x\},$$

$$Y_0 = \{w \in \widehat{B}(u, v) \text{ s.t. } \Phi_w \geq \Phi_y\},$$

$$x_0 = \arg \max\{\widehat{\Phi}_w : w \in X_0\},$$

(breaking ties arbitrarily) and similarly

$$y_0 = \arg \min\{\widehat{\Phi}_w : w \in Y_0\}.$$

Note that by assumption the pair X_0, Y_0 forms a partition of $\widehat{B}(u, v)$. By assumption,

$$\Phi_{x_0} \leq \Phi_x < \Phi_y - 4\tau \leq \Phi_{y_0} - 4\tau,$$

which implies for all $x' \in X_0$ and $y' \in Y_0$

$$\widehat{\Phi}_{y'} \geq \widehat{\Phi}_{y_0} > \widehat{\Phi}_{x_0} + 4\tau - 2\tau \geq \widehat{\Phi}_{x_0} + 2\tau \geq \widehat{\Phi}_{x'} + 2\tau.$$

Therefore, we have $x < y$. ■

Proposition 5 (Mini Reconstruction) *Let u, v be as above. Assume that $2m + 3\tau < M$. Then we have the following:*

1. [Reconstructed Edges Are Correct] *For each $j = 1, \dots, r(u, v)$, there is a unique edge e in $E^{(i)}$ such that*

$$b_{T^{(i)}}(e) \cap \widehat{B}(u, v) = \psi_j(u, v),$$

where the intersection on the left is applied separately to each set in the partition;

2. [Long Edges Are Present] *Let $e \in E^{(i)}$ with $e \in P_{T^{(i)}}(u, v)$ and $\lambda_e^{(i)} > 4\tau$. Then there is a unique j such that*

$$b_{T^{(i)}}(e) \cap \widehat{B}(u, v) = \psi_j(u, v).$$

Proof: Part 1 follows from Proposition 3 and Proposition 4 Part 2. Indeed, by Proposition 4 Part 2, $\psi_j(u, v)$ is a correct bipartition of $T^{(i)}$ restricted to $\widehat{B}(u, v)$. It corresponds to a unique edge of the latter tree because it is a full bipartition of $\widehat{B}(u, v)$. By Proposition 3, every edge of $T^{(i)}$ is witnessed in $\widehat{B}(u, v)$, so $\psi_j(u, v)$ must also correspond to a unique edge in $T^{(i)}$.

Similarly, Part 2 follows from Proposition 3 and Proposition 4 Parts 2 and 3. ■

3.3 Extending bipartitions: Reconstructing the components

Let $u, v \in \hat{v}_m^{(i)}$ with $(u, v) \in \hat{e}_m^{(i)}$ and let $\psi_j(u, v)$ be one of the bipartitions returned by MINI CONTRACTOR when given $(\hat{h}_m^{(i)}; u, v)$ as input. Let $e = (x, y) \in E^{(i)}$ be the edge of $T^{(i)}$ corresponding to $\psi_j(u, v)$ (as guaranteed by Proposition 5) and denote its bipartition by

$$b_{T^{(i)}}(e) = \{b^{(u)}, b^{(v)}\},$$

where $b^{(u)}$ and $b^{(v)}$ are respectively the sides containing u and v .

Proposition 6 (Leaves Outside Ball) *Assume that $2m + 3\tau < M$. Let $w \in \hat{v}_m^{(i)} - \widehat{B}(u, v)$. Assume that $w \in b^{(v)}$. Then, for all leaves w' in $b^{(u)}$ we have*

$$\hat{d}(w, w') \geq m.$$

Proof: Assume by contradiction that there is $w' \in b^{(u)}$ such that $\hat{d}(w, w') < m$. The path between w and w' must go through e since w and w' are on different sides of the partition. Therefore, for one of the endpoints of e , say x , we have $d(w, x) < m + \tau$. Also, since $d(u, x) \leq d(u, v) < \hat{d}(u, v) + \tau < m + \tau$, we have

$$d(w, u) < d(w, x) + d(x, u) < 2m + 2\tau < M.$$

We finally get

$$\hat{d}(w, u) < d(w, u) + \tau < 2m + 3\tau < M,$$

and similarly for $\hat{d}(w, v)$, a contradiction since we assumed $w \notin \hat{B}(u, v)$. ■

Proposition 7 (Correct Extension) *The bipartition $\bar{\psi}_j(u, v)$ returned by EXTENDER is correct, that is, $\bar{\psi}_j(u, v) = b_{T^{(i)}}(e)$.*

Proof: Let $K, \psi_j^{(u)}(u, v), \psi_j^{(v)}(u, v)$ be as in Figure 7. Since $\hat{h}_m^{(i)}$ is connected and we only remove edges between $\psi_j^{(u)}(u, v)$ and $\psi_j^{(v)}(u, v)$ to form K , it follows from Proposition 6 that all vertices in $\hat{v}_m^{(i)} - (\psi_j^{(u)}(u, v) \cup \psi_j^{(v)}(u, v))$ are connected in K to either $\psi_j^{(u)}(u, v)$ or $\psi_j^{(v)}(u, v)$. ■

We finally get the following.

Proposition 8 (Correctness of Main Loop) *Let $\{\hat{T}^{(i)}\}_{i=1}^q$ be the trees obtained at the end of the Main Loop of our algorithm. Then, for all $i = 1, \dots, q$, $\hat{T}^{(i)}$ is a refinement of $F_{4\tau, +\infty}(T^{(i)})$.*

Proof: By Propositions 5 and 7, all reconstructed edges are correct and they include at least those edges longer than 4τ . ■

3.4 Path-disjointness: Length and depth of shared edges

Let $T^{(i_1)}, T^{(i_2)}$ be the tree T restricted to components $\hat{h}_m^{(i_1)}, \hat{h}_m^{(i_2)}$ respectively. Note that each edge in $T^{(i_j)}$ is actually a path in T .

Proposition 9 (Path-Disjointness) *For all $u_1, v_1 \in L^{(i_1)}$ and $u_2, v_2 \in L^{(i_2)}$ such that*

$$\tilde{P}_T(u_1, v_1) \cap \tilde{P}_T(u_2, v_2) \neq \emptyset,$$

it holds that

1. [Depth of Shared Vertices] *We have*

$$\min\{\Delta_v(z) : z \in \tilde{P}_T(u_1, v_1) \cap \tilde{P}_T(u_2, v_2)\} \geq \frac{1}{2}(m - 3\tau).$$

2. [Length of Shared Edges] *If, further, $P_T(u_1, v_1) \cap P_T(u_2, v_2) \neq \emptyset$ then*

$$\max\{\lambda_e : e \in P_T(u_1, v_1) \cap P_T(u_2, v_2)\} \leq 2\tau.$$

Proof: Let $z \in \tilde{P}_T(u_1, v_1) \cap \tilde{P}_T(u_2, v_2)$. For $j = 1, 2$, by Proposition 2, there are leaves x_j, y_j in $L^{(i_j)}$ such that $z \in \tilde{P}_{T^{(i_j)}}(x_j, y_j)$ and $d(x_j, y_j) < m + \tau$.

For Part 1, assume without loss of generality that $d(x_2, z) < \frac{1}{2}(m + \tau)$. Then, for all $w \in L^{(i_1)}$,

$$\begin{aligned} d(w, z) &\geq d(w, x_2) - d(z, x_2) \\ &\geq m - \tau - \frac{1}{2}(m + \tau) \\ &\geq \frac{1}{2}(m - 3\tau). \end{aligned}$$

A similar argument applies to $w \in L^{(i_2)}$ and $w \in L - (L^{(i_1)} \cup L^{(i_2)})$.

For Part 2, let $e = (x, y) \in P_T(u_1, v_1) \cap P_T(u_2, v_2)$. Assume without loss of generality that the path from x to y partitions $\{x_1, y_1, x_2, y_2\}$ as $\{\{x_1, x_2\}, \{y_1, y_2\}\}$ in T , where x_1, x_2, y_1, y_2 were defined above. We have

$$\begin{aligned} 2d(x, y) &= d(x_1, y_1) + d(x_2, y_2) - d(x_1, x_2) - d(y_1, y_2) \\ &< \hat{d}(x_1, y_1) + \hat{d}(x_2, y_2) - \hat{d}(x_1, x_2) - \hat{d}(y_1, y_2) + 4\tau \\ &< 2m - 2m + 4\tau \\ &< 4\tau, \end{aligned}$$

where the third line follows from the definition of the clustering graph \hat{H}_m . ■

3.5 Proof of Main Theorem

Proof of Theorem 1: Part 1 follows from Proposition 9. Recall that

$$m < \frac{1}{2}[M - 3\tau].$$

Part 2 then follows from Proposition 8 and Proposition 1. ■

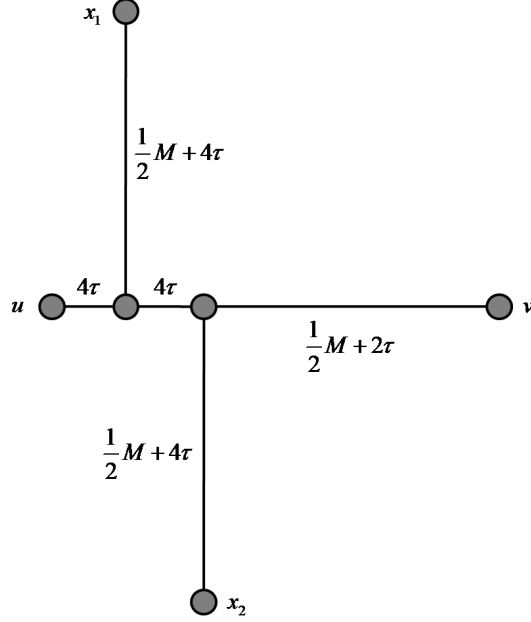


Figure 8: Counter-example: Reference tree T_0 .

4 Tightness of the Result

We showed that given a (τ, M) -distortion we reconstruct a subforest of T with chord depth $\approx \frac{1}{2}M$ which includes all edges of length at least 4τ . It may seem that we are losing a factor 2 in the chord depth and that, in fact, we should be able to reconstruct edges of chord depth close to M . But this is not the case. We show in this section that the chord depth of $\approx \frac{1}{2}M$ is essentially best possible (up to $O(\tau)$).

Consider the tree T_0 depicted in Figure 8. The tree T_0 has four leaves u, v, x_1, x_2 with adjacent edges of length respectively 4τ , $\frac{1}{2}M + 2\tau$, $\frac{1}{2}M + 4\tau$, and $\frac{1}{2}M + 4\tau$. The middle edge has length 4τ and the corresponding bipartition is $\{\{u, x_1\}, \{v, x_2\}\}$. Assume that we have the following (τ, M) -distortion of the metric corresponding to T_0 :

$$\begin{aligned} \hat{d}_0(u, v) &= \frac{1}{2}M + 10\tau, \quad \hat{d}_0(u, x_1) = \frac{1}{2}M + 8\tau, \quad \hat{d}_0(u, x_2) = \frac{1}{2}M + 12\tau, \\ \hat{d}_0(v, x_1) &= \hat{d}_0(v, x_2) = \hat{d}_0(x_1, x_2) = +\infty. \end{aligned}$$

Now, note that \hat{d}_0 is also a (τ, M) -distortion for the tree T_1 depicted

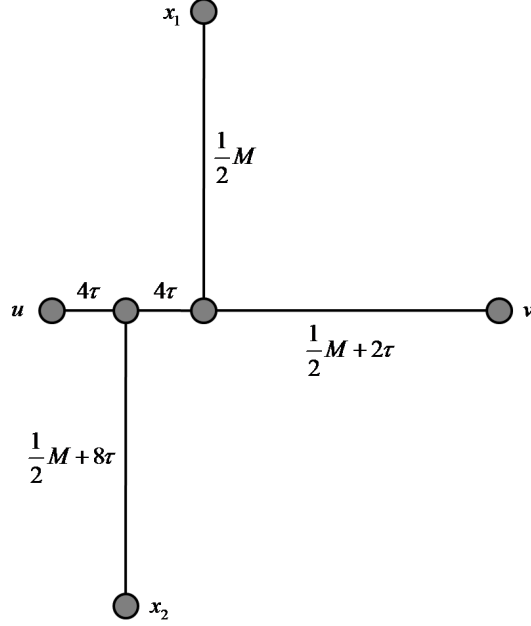


Figure 9: Counter-example: Tree T_1 with equivalent distortion.

in Figure 9. The tree T_1 has four leaves u, v, x_1, x_2 with adjacent edges of length respectively 4τ , $\frac{1}{2}M + 2\tau$, $\frac{1}{2}M$, and $\frac{1}{2}M + 8\tau$. The middle edge has length 4τ and the corresponding bipartition is $\{\{u, x_2\}, \{v, x_1\}\}$.

Hence, the two incompatible trees T_0 and T_1 cannot in general be distinguished from a (τ, M) -distortion. In particular, note that the middle edge of T_0 has length 4τ and chord depth $\frac{1}{2}M + 10\tau$, yet its bipartition cannot be recovered. This proves the claim.

5 Implementation

We briefly discuss the running time of the algorithm.

Building the graph \widehat{H}_m takes time $O(n^2)$, since we have to consider all pairs of leaves, and we find the connected components of \widehat{H}_m with Breadth-First-Search in another $O(n^2)$. We argue next that, for $i = 1, \dots, q$, we need $O(n_i^5)$ to build $\widehat{T}^{(i)}$, where $n_i = |\widehat{v}_m^{(i)}|$. We show first that for all pairs of leaves u and v , MINI CONTRACTOR and EXTENDER take time $O(n_i^3)$. Indeed, MINI CONTRACTOR takes time $O(n_i)$, since its running time is

linear in the size of $\widehat{B}(u, v)$; and EXTENDER takes time $O(n_i^3)$, since for each bipartition $\psi_j(u, v)$ —there are at most $O(n_i)$ of those—it is enough to perform a BFS. Given all bipartitions of the tree $\widehat{T}^{(i)}$, we use the standard TREE POPPING algorithm of [Mea81, BD86] to build $\widehat{T}^{(i)}$; since we have $O(n_i^3)$ bipartitions (not all of them distinct) this last step takes time $O(n_i^4)$. So for each tree i we need $O(n_i^5)$, and summing over i 's the total running time becomes $O(n^5)$.

We can improve on this running time by a more efficient implementation of EXTENDER as follows. For all $j = 0, \dots, r(u, v)$, we remove from the graph $\widehat{h}_m^{(i)}$ all leaves in $\cup_{\ell \neq j} C_\ell$ and perform a Breadth-First-Search to discover the leaves $K_j \subset \widehat{v}_m^{(i)} \setminus \widehat{B}(u, v)$ reachable in $\widehat{h}_m^{(i)}$ from the leaves in C_j . From an easy modification of Propositions 5 and 6, it follows that for every $w \notin \widehat{B}(u, v)$ there is at most one $j \in \{0, \dots, r(u, v)\}$ such that w is connected to a leaf in C_j . Given this, we can argue that we can recover the bipartitions $\bar{\psi}_j(u, v)$, $j = 1, \dots, r(u, v)$ from the K_j 's. The overall time needed by the BFS's is $O(n_i^2)$, hence $\widehat{T}^{(i)}$ can be computed in time $O(n_i^4)$ and our total running time becomes $O(n^4)$.

The above implementation is wasteful in running a BFS for every pair of leaves u and v with the possibility of creating as many as $O(n^3)$ bipartitions, each requiring $O(n)$ storage. Note that there are in fact at most n distinct bipartitions in T . To improve on the running time one may need to combine the BFS's performed in the above implementation by interleaving the MINI CONTRACTOR and EXTENDER steps with the TREE POPPING algorithm.

6 Concluding remarks

An interesting question for future work is whether the *approximate* disjointness in our results can be avoided. Since we guarantee that any shared edge lies deep inside the forest, it is tempting to simply remove all deep edges (say beyond $m/4$) from the output forest. Unfortunately, many of these edges may in fact be contracted and moreover they may be clustered in “supernodes” including both deep and not-so-deep edges. It does not seem to be a trivial task to break these deep supernodes apart and preserve strong reconstruction guarantees.

References

- [Att99] K. Atteson. The performance of neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*, 25(2-3):251–278, 1999.
- [BD86] Hans-Jürgen Bandelt and Andreas Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Adv. in Appl. Math.*, 7(3):309–343, 1986.
- [BH87] Daniel Barry and J. A. Hartigan. Statistical analysis of hominoid molecular evolution. *Statist. Sci.*, 2(2):191–210, 1987. With comments by Stephen Portnoy and Joseph Felsenstein and a reply by the authors.
- [Bun71] P. Buneman. The recovery of trees from measures of dissimilarity. In *Mathematics in the Archaeological and Historical Sciences*, pages 187–395. Edinburgh University Press, Edinburgh, 1971.
- [CDvM⁺06] Francesca D. Ciccarelli, Tobias Doerks, Christian von Mering, Christopher J. Creevey, Berend Snel, and Peer Bork. Toward Automatic Reconstruction of a Highly Resolved Tree of Life. *Science*, 311(5765):1283–1287, 2006.
- [Cha96] Joseph T. Chang. Full reconstruction of Markov models on evolutionary trees: identifiability and consistency. *Math. Biosci.*, 137(1):51–73, 1996.
- [CK01] Miklós Csurös and Ming-Yang Kao. Provably fast and accurate recovery of evolutionary trees through harmonic greedy triplets. *SIAM Journal on Computing*, 31(1):306–322, 2001.
- [Csu02] M. Csurös. Fast recovery of evolutionary trees with thousands of nodes. *J. Comput. Biol.*, 9(2):277–97, 2002.
- [CT06] Benny Chor and Tamir Tuller. Finding a maximum likelihood tree is hard. *J. ACM*, 53(5):722–744, 2006.
- [Day87] William H. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bull. Math. Biol.*, 49(4):461–467, 1987.

- [DHJ⁺06] Constantinos Daskalakis, Cameron Hill, Alexander Jaffe, Radu Mihaescu, Elchanan Mossel, and Satish Rao. Maximal accurate forests from distance matrices. In *RECOMB*, pages 281–295, 2006.
- [DMR06] Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Optimal phylogenetic reconstruction. In *STOC’06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 159–168, New York, 2006. ACM.
- [DMR09] Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Phylogenies without branch bounds: Contracting the short, pruning the deep. To appear in RECOMB 2009, 2009.
- [DS86] William H. E. Day and David Sankoff. Computational complexity of inferring phylogenies by compatibility. *Syst. Zool.*, 35(2):224–229, 1986.
- [ESSW99a] P. L. Erdős, M. A. Steel, L. A. Székely, and T. A. Warnow. A few logs suffice to build (almost) all trees (part 1). *Random Struct. Algor.*, 14(2):153–184, 1999.
- [ESSW99b] P. L. Erdős, M. A. Steel, L. A. Székely, and T. A. Warnow. A few logs suffice to build (almost) all trees (part 2). *Theor. Comput. Sci.*, 221:77–118, 1999.
- [Fel78] J. Felsenstein. Cases in which parsimony or compatibility methods will be positively misleading. *Syst. Biol.*, pages 401–410, 1978.
- [Fel04] J. Felsenstein. *Inferring Phylogenies*. Sinauer, Sunderland, MA, 2004.
- [GF82] R. L. Graham. and L. R. Foulds. Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computational time. *Math. Biosci.*, 60:133–142, 1982.
- [GMS08] Ilan Gronau, Shlomo Moran, and Sagi Snir. Fast and reliable reconstruction of phylogenetic trees with very short edges. To appear in SODA, 2008.

- [HNW99] D. H. Huson, S. H. Nettles, and T. J. Warnow. Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *J. Comput. Biol.*, 6(3–4), 1999.
- [KZZ03] Valerie King, Li Zhang, and Yunhong Zhou. On the complexity of distance-based evolutionary tree reconstruction. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 444–453, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.
- [Lak94] JA Lake. Reconstructing Evolutionary Trees from DNA and Protein Sequences: Paralinear Distances. *Proceedings of the National Academy of Sciences*, 91(4):1455–1459, 1994.
- [LC06] Michelle R. Lacey and Joseph T. Chang. A signal-to-noise analysis of phylogeny estimation by neighbor-joining: insufficiency of polynomial length sequences. *Math. Biosci.*, 199(2):188–215, 2006.
- [LSHP94] PJ Lockhart, MA Steel, MD Hendy, and D Penny. Recovering Evolutionary Trees under a More Realistic Model of Sequence. *Mol Biol Evol*, 11(4):605–612, 1994.
- [Mea81] C. A. Meacham. A manual method for character compatibility analysis. *Taxon*, 30:591–600, 1981.
- [Mos07] E. Mossel. Distorted metrics on trees and phylogenetic forests. *IEEE/ACM Trans. Comput. Bio. Bioinform.*, 4(1):108–116, 2007.
- [MR06] Elchanan Mossel and Sébastien Roch. Learning nonsingular phylogenies and hidden Markov models. *Ann. Appl. Probab.*, 16(2):583–614, 2006.
- [PL98] H. Philippe and J. Laurent. How good are deep phylogenetic trees? *Current Opinion in Genetics & Development*, 8:616–623(8), December 1998.
- [Roc06] Sébastien Roch. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 3(1):92–94, 2006.

- [SN87] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–425, 1987.
- [SS99] Michael A. Steel and László A. Székely. Inverting random functions. *Ann. Comb.*, 3(1):103–113, 1999. Combinatorics and biology (Los Alamos, NM, 1998).
- [SS02] M. A. Steel and L. A. Székely. Inverting random functions. II. Explicit bounds for discrete maximum likelihood estimation, with applications. *SIAM J. Discrete Math.*, 15(4):562–575 (electronic), 2002.
- [SS03] C. Semple and M. Steel. *Phylogenetics*, volume 22 of *Mathematics and its Applications series*. Oxford University Press, 2003.
- [Ste94] M. Steel. Recovering a tree from the leaf colourations it generates under a Markov model. *Appl. Math. Lett.*, 7(2):19–23, 1994.

A Log-Det Estimator

For completeness, we relate the definition of the distorted metric (see Definition 1) to its biological context. In phylogenetic reconstruction, a distorted metric is naturally derived from samples of a Markov model on a tree—a common model of DNA sequence evolution used in Biology.

Definition 7 (Markov model on a tree) *A Markov model on a tree is the following stochastic process:*

- Let $T_\rho = (V, E, \rho)$ be a finite tree rooted at ρ . Denote by E_\downarrow the set E directed away from the root.
- Let $L = [n]$ be the leaf set of T_ρ .
- Let \mathcal{R} be a finite set with r elements.
- Associate to each edge $e \in E$ a $r \times r$ stochastic matrix $M(e)$ with $\det M(e) > 0$.
- Let π_ρ be a distribution on \mathcal{R} with $\pi_\rho(\sigma) > 0$ for all $\sigma \in \mathcal{R}$.

The process runs as follows. Pick a state for the root according to π_ρ . Moving away from the root toward the leaves, apply the channel $M(e)$ to each edge e independently. Denote the state so obtained $\sigma_V = (\sigma_v)_{v \in V}$. In particular, $\sigma_{[n]}$ is the state at the leaves. More precisely, the joint distribution of σ_V is given by

$$\mu_V(\sigma_V) = \pi_\rho(\sigma_\rho) \prod_{e=(x,y) \in E_\downarrow} (M(e))_{\sigma_x \sigma_y},$$

and therefore the distribution at the leaves is

$$\mu_L(\sigma_L) = \sum_{\sigma'_V: \sigma'_L = \sigma_L} \pi_\rho(\sigma'_\rho) \prod_{e=(x,y) \in E_\downarrow} (M(e))_{\sigma'_x \sigma'_y}.$$

For $W \subseteq V$, we denote by μ_W the marginal of μ_V at W .

More generally, we are given k independent samples $(\sigma_{[n]}^i)_{i=1}^k$ from the same Markov model. We think of $(\sigma_l^i)_{i=1}^k$ as the sequence at $l \in [n]$. Typically in biological applications $\mathcal{R} = \{A, G, C, T\}$. MMTs model how DNA sequences stochastically evolve by point mutations along an evolutionary tree—under the assumption that each site in the sequences evolves independently.

In the phylogenetic reconstruction problem, we are given sequences $(\sigma_{[n]}^i)_{i=1}^k$ (one sequence for each extant species) and our goal is to recover the generating tree—or more precisely its unrooted version (the root is typically not identifiable [Ste94]). A natural place to start is to measure a notion of “distance” between the leaves. That is, we seek to associate to an MMT an additive metric as defined in Definition 8. In general, this can be achieved using the so-called log-det distance.

Definition 8 (Log-Det Distance [Ste94]. See also [BH87, LSHP94, Lak94].) Consider the Markov model in Definition 7. Associate to each edge $e = (u, v) \in E_\downarrow$ a weight $\lambda(e)$ as follows:

- If e is a leaf edge then

$$\lambda(e) = -\log \det M(e) - \frac{1}{2} \log \prod_{\sigma' \in \mathcal{R}} \mu_u(\sigma').$$

- Otherwise

$$\lambda(e) = -\log \det M(e) - \frac{1}{2} \log \prod_{\sigma' \in \mathcal{R}} \mu_u(\sigma') + \frac{1}{2} \log \prod_{\sigma' \in \mathcal{R}} \mu_v(\sigma').$$

The log-det distance is defined as: $\forall u, v \in L$,

$$d(u, v) \equiv -\log \det F(u, v) = \sum_{e \in P_T(u, v)} \lambda_e,$$

where

$$\forall \sigma', \sigma'' \in \mathcal{R}, (F(u, v))_{\sigma', \sigma''} = \mu_{\{u, v\}}(\sigma_u = \sigma', \sigma_v = \sigma'').$$

It was shown in [Ste94] that the log-det distance is indeed an additive metric.

When the sequence length k is finite, we can only obtain an estimate \hat{d} of d

$$\hat{d}(u, v) = -\log \det \hat{F}(u, v),$$

where

$$\forall \sigma', \sigma'' \in \mathcal{R}, (\hat{F}(u, v))_{\sigma', \sigma''} = \frac{1}{k} \sum_{i=1}^k \mathbb{1}\{\sigma_u^i = \sigma', \sigma_v^i = \sigma''\}.$$

The next lemma, a slight generalization of Proposition 2.1 in [Mos07], shows that such an estimator constitutes a *distorted metric*.

Lemma 1 (Log-Det Distance: Distorted Metric) *Let \hat{d} be the estimator defined above. Then there is a constant $\Lambda > 0$ such that if one chooses (τ, M) with*

$$k \geq \frac{\Lambda}{(1 - e^{-\tau})^2} e^{2M+4\tau} \log n,$$

then \hat{d} is a (τ, M) -distortion with probability $1 - 1/\text{poly}(n)$.

Proof: Fix $u, v \in L$. Denote $F = F(u, v)$, $\hat{F} = \hat{F}(u, v)$, $\omega = d(u, v)$, and $\hat{\omega} = \hat{d}(u, v)$. We assume that k is at least $\Omega(\log n)$. Let \hat{F}' be \hat{F} with one sample arbitrarily changed. It was argued in [ESSW99b] that there are constants c_1, c_2 such that

$$\left| \det \hat{F} - \det \hat{F}' \right| \leq \frac{c_1}{k},$$

and

$$\left| \det F - \mathbb{E}[\det \hat{F}] \right| \leq \frac{c_2}{k}.$$

Assume $\omega < M + 2\tau$ (in particular if $\omega < M + \tau$). By Azuma's inequality,

$$\begin{aligned} \mathbb{P}[\hat{\omega} \geq \omega + \tau] &= \mathbb{P}[\det \hat{F} - \det F \leq -(\det F)(1 - e^{-\tau})] \\ &\leq \mathbb{P}\left[\det \hat{F} - \mathbb{E}[\det \hat{F}] \leq e^{-M-2\tau}(1 - e^{-\tau}) - \frac{c_2}{k}\right] \\ &\leq \exp\left(-\frac{k}{2c_1^2} \left(e^{-M-2\tau}(1 - e^{-\tau}) - \frac{c_2}{k}\right)^2\right), \end{aligned}$$

where we assume k is large enough that

$$e^{-M-2\tau}(1 - e^{-\tau}) - \frac{c_2}{k} \geq 0.$$

The same inequality holds for $\mathbb{P}[\hat{\omega} \leq \omega - \tau]$.

On the other hand, assume $\omega > M + 2\tau$. Then,

$$\begin{aligned} \mathbb{P}[\hat{\omega} \leq M + \tau] &\leq \mathbb{P}\left[\det \hat{F} - \mathbb{E}[\det \hat{F}] \geq e^{-M-\tau} - e^{-M-2\tau} - \frac{c_2}{k}\right] \\ &\leq \exp\left(-\frac{k}{2c_1^2} \left(e^{-M-\tau}(1 - e^{-\tau}) - \frac{c_2}{k}\right)^2\right). \end{aligned}$$

■